UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/722,630	11/26/2003	Brent Roberts	717841.5	7332

27128 7590 03/09/2009
HUSCH BLACKWELL SANDERS LLP
720 OLIVE STREET
SUITE 2400
ST. LOUIS, MO 63101

EXAMINER	
NUNEZ, JORDANY	
ART UNIT	**PAPER NUMBER**
2175	
NOTIFICATION DATE	**DELIVERY MODE**
03/09/2009	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

pto-sl@huschblackwell.com

Office Action Summary	Application No.	Applicant(s)	
	10/722,630	ROBERTS ET AL.	
	Examiner	Art Unit	
	Jordany Núñez	2175	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *22 December 2008*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-17* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-17* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)	4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)	5) ☐ Notice of Informal Patent Application (PTO-152)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.	6) ☐ Other: _____.

DETAILED ACTION

Continued Examination Under 37 CFR 1.114

A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on 12/22/2008 has been entered.

Claim Rejections - 35 USC § 102

1. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

2. Claims 1-17 are rejected under 35 U.S.C. 102(b) as being anticipated by Sanderson (20020101448).

Re claim 1, Sanderson discloses an integrated system of frameworks and data repositories for generating a graphical user interface in a client-server environment comprising: a user interface (UI) repository (data factory 210 for example) residing in a database accessible to a client server hardware system, where said UI repository, contains a UI element (data element 208 for example), which defines data element attributes including data type, how to display data and labels (see figure 2 for example); a screen repository (specification 209 for example) residing in the database storage device accessible to the client-server hardware system, where said screen repository includes screen attributes, which defines the hierarchical navigational tree structure of screens for an graphical user interface (GUI) application and further defines what screen will be constructed and defines a GUI component of the screen based on the data type (workflow data, configuration data, see abstract for example); a data binding framework (content factory 213 for example) operable to bind data to the UI element and the GUI component based on data type (inherent in order to display); a (GUI) framework (controller 201c for example) operably residing at a client in the client-server hardware system, where said (GUI) framework is operable to control how data is handled and processed within the GUI component of the GUI application based on data type including binding data to the GUI component utilizing the data binding framework; and

a navigation framework (view 201b, model 201a for example) operably residing at the client, where said navigation framework controls generating and displaying of the screens within an application and further builds a navigation tree structure based on the screen attributes; and

an object oriented software application (see paragraph 0037, object-oriented implementation and XML, see paragraph 0046 for example) for generating a graphical user interface being executed in said client-server hardware system (client 201D, server 206 in figure 2 for example) having functional interfaces for accessing repositories and frameworks accessible by the client-server network (see figure 2 for example).

Re claim 2, Sanderson discloses an integrated system of frameworks and data repositories for generating a graphical user interface, further comprising: a security framework (using 442 for example, inherent) operable to communicate information to the navigation framework causing the navigation framework to selectively deny a user access to screens by not providing the user with selections that would navigate to the screen (see figure 4b).

Re claim 3, Sanderson discloses an integrated system of frameworks and data repositories for generating a graphical user interface, further comprising: a verification framework (validator 211 for example) operable to apply business rules to data contained in a data set and determine if the data is in error and if in error the verification framework communicates with the GUI component to display an error message.

Re claim 4, Sanderson discloses an integrated system of frameworks and data repositories for generating a graphical user interface in a client-server environment comprising:

a graphical user interface (GUI) framework (201c for example) operably residing at a client server hardware system in a client-server network environment, where said GUI framework is operable to control how data is handled and processed within a GUI component of a GUI application based on a data type including binding data to the GUI component utilizing a data binding framework;

a collection of integrated repositories (210 for example) relationally inter referenced by UI elements defining the data type within their respective attribute tables operable for accessing and integrating all attribute elements relating to generating a graphical user interface;

a collection of executable object oriented routines (209 for example) being executed on the client-server hardware system (client 201D, server 206, see figure 2 for example) and operable to manipulate the GUI framework;

a collection of XML files (content 207 for example) operable to access and export data from the repositories at run time for use by the GUI application; an XML layout manager (format 212 for example) operable to define the screen layout from the collection of XML files; and a navigation framework operably residing at the client, where said navigation framework (view 201b for example) controls generating and displaying of screens within the GUI application based upon the XML lay out manager and the repository attributes accessed and exported by the collection of XML files and further builds the navigation tree structure based on the repository attributes.

Re claim 5, Sanderson discloses an integrated system of frameworks and data repositories for generating a graphical user interface, further comprising: a security framework (using 442, inherent) operable to communicate information to the navigation framework causing the navigation framework to selectively deny a user access to screens by not providing the user with selections that would navigate to the screen (see figure 4b).

Re claim 6, Sanderson discloses an integrated system of frameworks and data repositories for generating a graphical user interface, further comprising: a verification framework (validator 211 for example) operable to apply business rules to data contained in a data set and determine if the data is in error and if in error the verification framework communicates with the GUI component to display an error message.

Re claim 7, Sanderson discloses an integrated system of frameworks and data repositories for generating a graphical user interface in a client-server environment comprising:

a screen repository (209 for example) residing in a database storage device accessible by a client-server hardware system, where said screen repository includes screen attributes, which defines the hierarchical navigational tree structure of screens for an graphical user interface (GUI) application and further defines what screen will be constructed and defines a GUI components of the screen;

a user interface (UI) repository (210 for example) residing in the database storage device accessible to the client server hardware system, where said UI repository, contains a UI element, which defines data element attributes including data type, how to display data and labels; and

a data binding framework (213 for example) operable to bind data to the UI element and the GUI component based on the data type defined in the UI repository; and

an object oriented software application (see paragraph 0037 object oriented implementation and XML in paragraph 0046 for example) for generating a graphical user interface being executed in said client-server hardware system (client 201D, server 206, see figure 2 for example) having functional interfaces for accessing repositories and frameworks accessible by the client-server hardware system (see figure 2 for example).

Re claim 8, Sanderson discloses an integrated system of frameworks and data repositories for generating a graphical user interface, further comprising: a security framework (using 442, inherent) operable to communicate information to the navigation framework causing the navigation framework to selectively deny a user access to screens by not providing the user with selections that would navigate to the screen (see figure 4b for example).

Re claim 9, Sanderson discloses an integrated system of frameworks and data repositories for generating a graphical user interface, further comprising: a verification framework (211 for example) operable to apply business rules to data contained in a data set and determine if the data is in error and if in error the verification framework communicates with the GUI component to display an error message.

Re claim 10, Sanderson discloses an integrated system tool comprising:

A computer executable administrative computing tool application electronically stored in electronic memory of a computer (server 206 for example) where said administrative computing tool includes, a navigation tool (201b for example) for building a navigation framework adapted to control the generation of screens for a graphical user interface (GUI) application and further adapted to define the hierarchical relationship of the screens, a screen repository tool (209 for example) operable to build a repository of screen attributes to establish a hierarchical screen navigation structure and a corresponding Java class construct to be executed and a GUI component, a user interface repository tool (210 for example) operable to build a user interface repository having user interface attribute tables of user interface elements corresponding to the GUI component, and a data binding framework tool (213 for example) operable to build a data binding framework operable to bind data from an appropriate data set to the user interface element and the GUI component,

Where said computer is operable to generate and display a graphical user interface utilizing said framework and said repository tools (see figure 2, items 201a and b for example).

Note that in regards to “in a test environment", it has been held that a recitation with respect to the manner in which a claimed apparatus is intended to be employed does not differentiate the claimed apparatus from a prior art apparatus satisfying the claimed structural limitations. *Ex parte Masham*, 2 USPQ2d 1647 (1987).

Re claim 11, Sanderson discloses an integrated system tool, where the administrator computing tool further comprises: a security framework tool (using 442 for example) operable to build a security framework operable to communicate information to the navigation framework causing the navigation framework to selectively deny a user access to screens by not providing the user with selections that would navigate to the screen (see figure 4b), where said computer is operable to generate and display a graphical user interface utilizing said framework and said repository tools (see figure 2, items 201a and b for example), additionally including said security framework tool..

Note that in regards to “in a test environment", it has been held that a recitation with respect to the manner in which a claimed apparatus is intended to be employed does not differentiate the claimed apparatus from a prior art apparatus satisfying the claimed structural limitations. *Ex parte Masham*, 2 USPQ2d 1647 (1987).

Re claim 12, Sanderson discloses an integrated system tool, where the administrator computing tool further comprises: a verification framework tool (211 for example) operable to build a verification framework operable to apply business rules to data contained in a data set and determine if the data is in error and if in error the verification framework communicates with the GUI component to display an error message, where said computer is operable to generate and display a graphical user interface utilizing said framework and said repository tools (see figure 2, items 201a and b for example), additionally including said verification framework tool..

Note that in regards to “in a test environment", it has been held that a recitation with respect to the manner in which a claimed apparatus is intended to be employed does not differentiate the claimed apparatus from a prior art apparatus satisfying the claimed structural limitations. *Ex parte Masham*, 2 USPQ2d 1647 (1987).

. Re claim 13, Sanderson discloses a method of generating a graphical user interface utilizing an integrated system of frameworks and data repositories comprising the steps of: Receiving a screen (201d to 207 for example) request to a graphical user interface (GUI) application based on a user input; Accessing and constructing (from 201b to 201a for example) a basic screen and screen attributes from a screen repository corresponding to the user input as determined by a navigation framework; Binding GUI components (create instance in 210 for example) defined by the screen attributes with user interface elements from a UI repository based on a data type defined in the UI repository (see figure 2 for example); Binding data to the GUI components and UI elements based on the data type (format 212 for example); and Displaying the screen on a display system (201b to 201a for example).

Re claim 14, Sanderson discloses a method of generating a graphical user interface, further comprising: filtering a screen (with 442 for example) with a security framework operable to communicate information to the navigation framework causing the navigation framework to selectively deny a user access to screens by not providing the user with selections that would navigate to the screen.

Re claim 15, Sanderson discloses a method of generating a graphical user interface as recited in claim 13, further comprising: displaying an error message (using 211 for example) with a verification framework operable to apply business rules to data contained in a data set and determine if the data is in error and if in error the verification framework communicates with the GUI component to display an error message.

Re claim 16, Sanderson discloses a method for building an integrated system of frameworks and data repositories for generating a graphical user interface comprising the steps of: building a graphical user interface (GUI) framework (201c for example) operable to reside at a client in a client-server network environment, where said GUI framework is operable to control how data is handled and processed within a GUI component of a GUI application including binding data to the GUI component utilizing a data binding framework; and

building a collection (with 210 for example) of integrated repositories to be relationally inter referenced by UI element data type within their respective attribute tables operable for accessing and integrating all attribute elements relating to generating a graphical user interface; and displaying a graphical user interface (see figure 2 and abstract for example).

Re claims 17, Sanderson discloses a method for building an integrated system further comprising the steps of: building a verification framework (with 211 for example) operable to apply business rules to data contained in a data set and determine if the data is in error and if in error the verification framework communicates with the GUI component to display an error message.

Response to Arguments

Applicant's arguments have been fully considered but are not persuasive. Examiner reiterates that references to specific columns, figures or lines should not be limiting in any way. The entire reference provides disclosure related to the claimed invention. Applicant argues that:

1) Sanderson teaches that the data factory 210 can include two types of data factories. One type can produce data elements suitable for transmission by the messenger function 202. Contents specification objects 209 are passed to the data factory, which specifies the structure and Symantecs thereby allowing this type of data factory to produce a suitable data element. The other type of data factory 210 can produce data elements suitable for interaction with widgets. Therefore, the data factory 210 is not a user interface repository containing UI elements of which define data element attributes including data type. To the contrary, the data factory 210 of Sanderson teaches the production of a data element based on transmitted content specifications. Therefore, the methodology is content driven and/or work task driven as opposed to a user interface repository driven by the UI elements residing therein.

Therefore, the intended purpose of the claimed invention is not accomplished by the Sanderson reference (page 10, last paragraph).

Examiner disagrees.

As acknowledged by Applicant, the data factory 210 of Sanderson teaches the production of a data element based on transmitted content specifications 209. Sanderson (page 6, paragraphs [0054] to [0056]) further that the data factory can be implemented as a dictionary of conventional factory objects keyed on a specification 209 which correlates to the same element. Furthermore, a widget factory uses specification 209 to determine which UI widget to produce to facilitate interaction between a user and a composite data element. Thus, Sanderson explicitly teaches the data factory 210 is a user interface repository (e.g., a dictionary of conventional objects specified by specification 209) containing UI elements of which define data element attributes including data type (e.g., determining which UI widget to use based on specification 209).

2) The content specification 209 is disclosed in Sanderson at paragraphs 48, 49 and 50. The content specification 209 as described in Sanderson is provided to specify the structure and semantics of data elements. Sanderson teaches that each content specification 209 can contain attributes that define the behavior and allowable structure of a data element. However, Sanderson does not anticipate, teach, or suggest a screen repository which includes the attributes of various screens and defines the hierarchical navigational tree structures of various screens and further defines what screen will be constructed (page 11, last paragraph).

Examiner disagrees.

Sanderson (page 6, paragraph 56) teaches using a specification 209 to determine which UI widgets to produce, and that the UI widgets include display panels and widgets for displaying and accepting user input for atomic data elements. Thus, Sanderson explicitly teaches a screen repository (e.g., displaying a specific display panel corresponding to a particular specification 209) which includes the attributes of various screens and defines the hierarchical navigational tree structures of various screens and further defines what screen will be constructed (e.g., a display panel displaying various UI widgets is a hierarchical navigational tree structure).

3) The Examiner further asserts that it is inherent for this to occur in order to display. The content factory taught by Sanderson is an object that generates content objects on behalf of a messenger 202. Therefore, the content factory does not teach the limitation relating to a data binding framework operable to bind data to the UI element and the GUI component based on data type (page 12, antepenultimate paragraph).

Examiner disagrees.

Sanderson (page 6, paragraph [0056] teaches determining a UI widget by using a specification 209 to determine which UI widget to produce. Sanderson (page 6, paragraph [0054] teaches a single data element 208 corresponding to a specification 209. Thus, Sanderson explicitly teaches a data binding framework operable to bind data to the UI element and the GUI component based on data type (e.g., a data element is manipulated by a UI widget because a UI widget is chosen based on specification 209).

4) The controller 201C of Sanderson is part of the task function 201, which is a self contained logical module that performs a specific kind of interaction with the user or otherwise a non-visual process. The controller portion of the self contained logical module is a function for initiating and responding to events associated with tasks 201. There is no discussion in Sanderson that the controller is operable to control how data is handled and processed within the GUI component (page 12, penultimate paragraph).

Examiner disagrees.

Sanderson (page 6, paragraph [0056] teaches determining a UI widget by using a specification 209 in order to facilitate interaction between a user and an individual or composite data element. Thus, Sanderson explicitly teaches a controller being operable to control how data is handled and processed within the GUI component (e.g., the UI generator controls which UI widgets are displayed to the user, based on specification 209, and the UI widgets, which cause communication with server 206, in turn control how data is handled and processed).

5) The Examiner also asserts that the data stream that is an XML document as described in paragraph 46 anticipates a software application for generating a graphical user interface having functional interfaces for accessing repositories and frameworks. However, the XML document referred to in paragraph 46 is merely content within a data stream (page 12, last paragraph).

Examiner disagrees.

In paragraph [0046] Sanderson teaches using an XML document, for example, to enable communications back and forth between a server and a declarative UI generator. Further, Sanderson (page 6, paragraph [0056] teaches determining a UI widget, which includes display panels and widgets, by using a specification 209 in order to facilitate interaction between a user and an individual or composite data element. Thus, Sanderson explicitly teaches software application for generating a graphical user interface (e.g., a declarative UI generator) having functional interfaces for accessing repositories and frameworks (e.g., defining display panels which facilitate manipulation of data elements by a user).

6) The data factory 210 as characterized in the remarks above with regard to independent claim 1 do not teach a collection of integrated repositories relationally inter-referenced by UI elements as recited in independent claim 4. Further, as argued for independent claim 1, the content 207 that are described as objects that can hold and translate data elements is not a collection of XML files operable to access and export data from repositories at one time and further the view function 201B does not anticipate the navigation framework as recited in independent claim 4 (page 13, penultimate paragraph).

Examiner disagrees.

As stated above, Sanderson teaches each and every limitation claimed. In particular, Sanderson teaches a collection of integrated repositories relationally inter-referenced by UI elements (e.g., data elements with corresponding specifications which are used to determine UI widgets which facilitate interaction between a user and data element); further, the content 207 are described as objects that can hold and translate data elements as a collection of XML files operable to access and export data from repositories at one time (e.g., XML is used to facilitate communications between a server and a declarative UI generator); finally, the view function 201B does anticipate the navigation framework (e.g., it is a UI driven by UI widgets, controlled by a server, and facilitation interaction between a user and a data element).

Conclusion

THIS ACTION IS MADE FINAL. Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Jordany Núñez whose telephone number is (571)272-2753. The examiner can normally be reached on Monday Through Thursday 9am-7:30pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner’s supervisor, William Bashore can be reached on (571)272-4088. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

JN
2/26/2009

/William L. Bashore/
Supervisory Patent Examiner, Art Unit 2175